TOWARDS A FLEXIBLE AND SECURE PYTHON PACKAGE REPOSITORY SERVICE

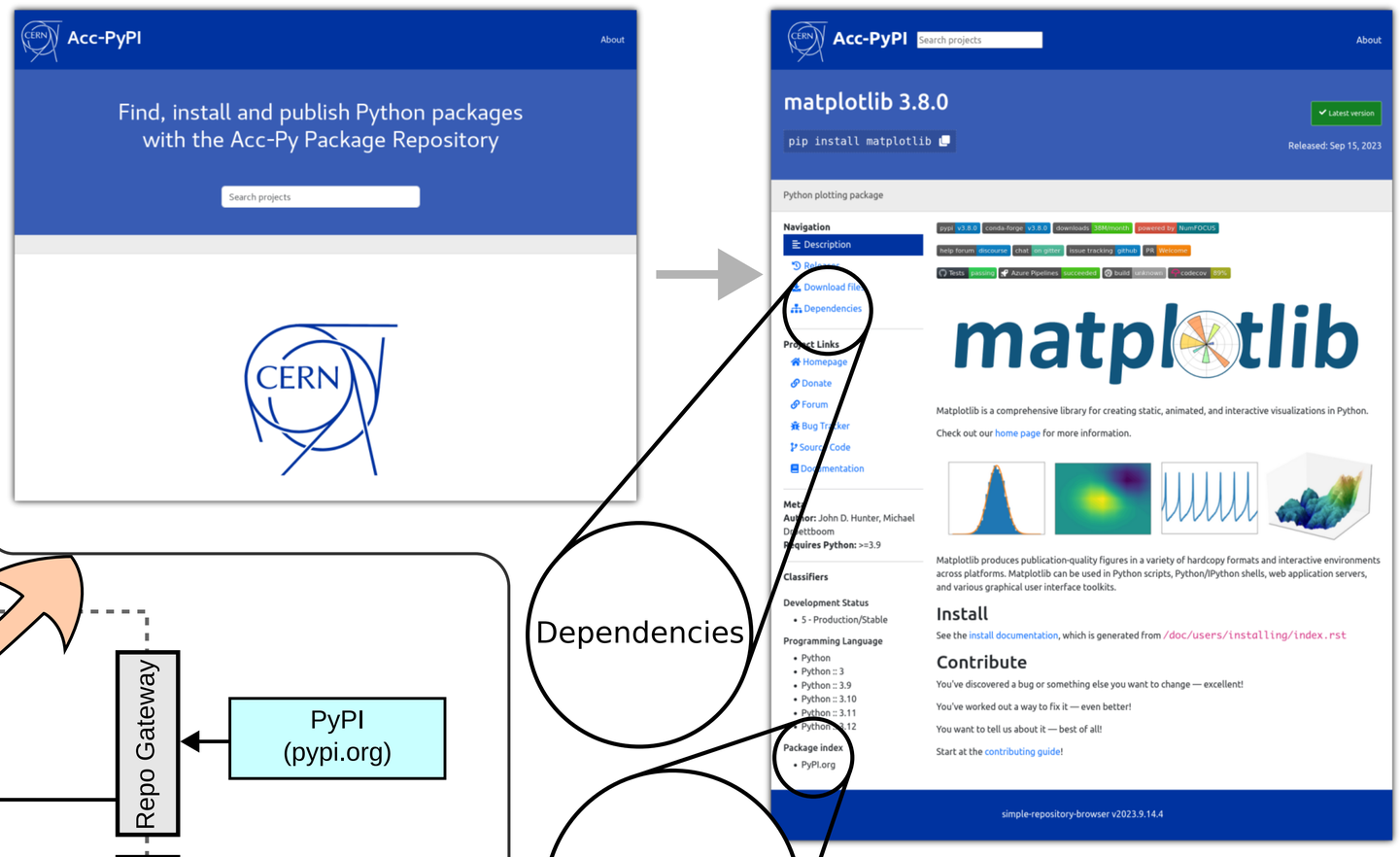
I. Sinkarenko, P. Elson, F. Iannaccone, W. Koorn, B.Copy, CERN, Geneva, Switzerland

Overview

The use of third-party and internal software packages has become a crucial part of modern software development. Despite its benefits, installing arbitrary software from a third-party package repository poses security and operational risks. For instance, the dependency confusion attack first published in 2021 has still not been fully addressed by the main open-source repository services. An in-house development was conducted to address this, using a modular approach to building a Python package repository, called "**Acc-Py Package Repository**", that enables the creation of a powerful and security-friendly repository service using small components. The solution is not CERN-specific and is likely to be relevant to other institutes facing comparable challenges.

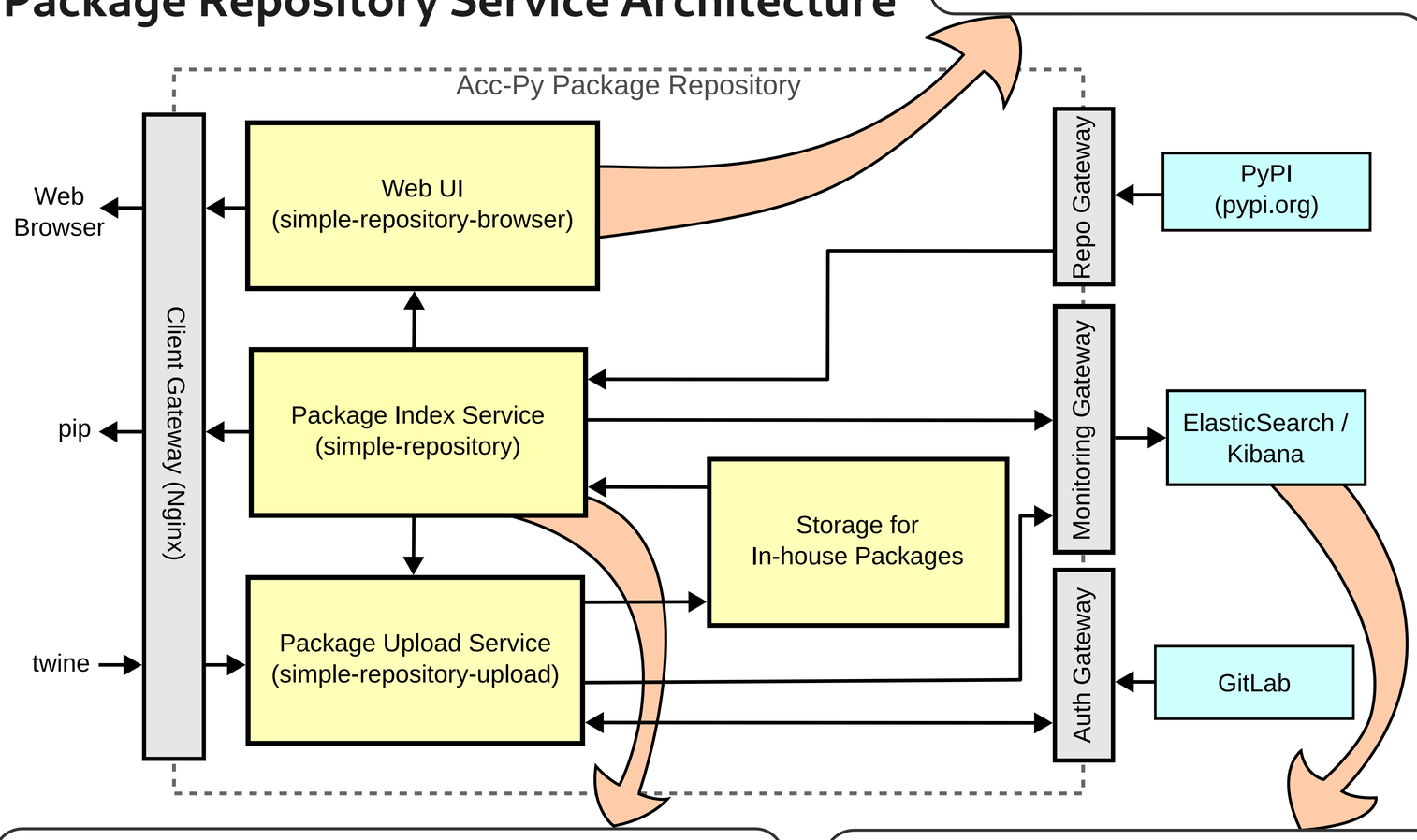
Web User Interface (Web UI)

- **Compatible with any Simple Repository API (PEP-503)**
- **Can be used separately as a standalone service**



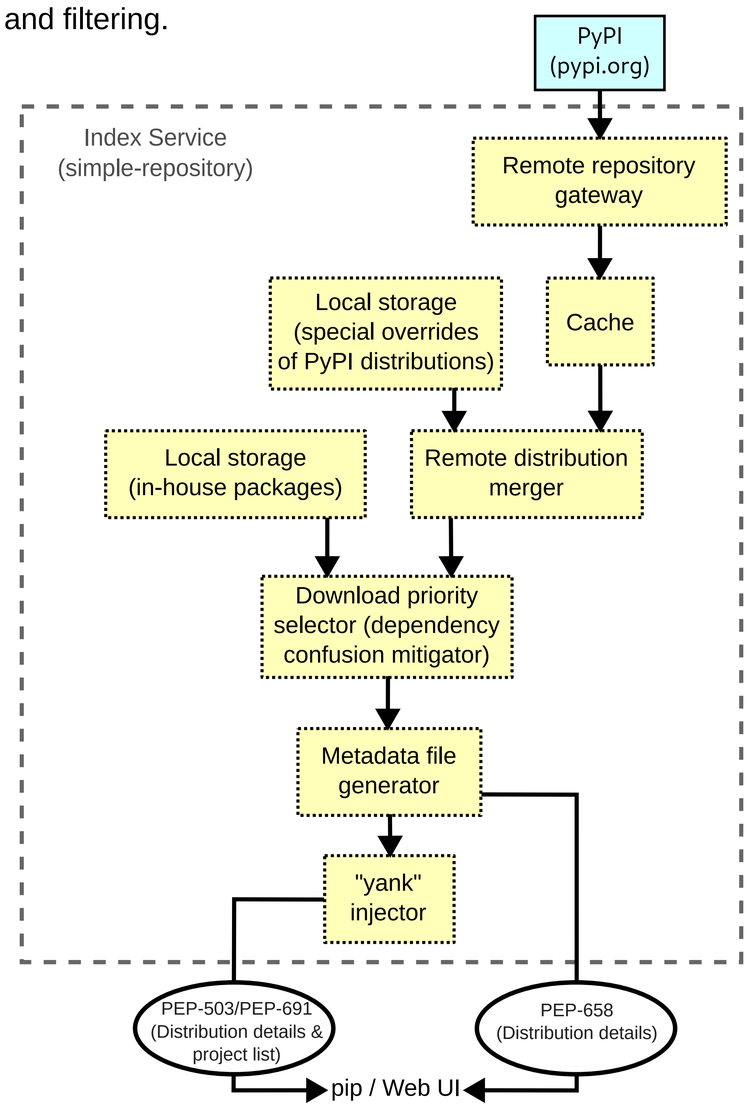
- Inspired by PyPI.org and is familiar to every Python developer
- Creates links to documentation, source code and other supporting information
- Can be configured to crawl and index all connected repositories for combined search capabilities
- Package details UI indicates in which of the repositories it has been found (PyPI or in-house)
- Presents additional information, e.g. package dependencies

Package Repository Service Architecture



Component-Based Approach Inside the Index Service

The implementation is simplified by the creation of reusable components that represent the abstraction of an upstream repository. Such an abstraction allows components to be chained together in the form of a directed acyclic graph, which progressively enhances repository data or performs operations such as caching and filtering.



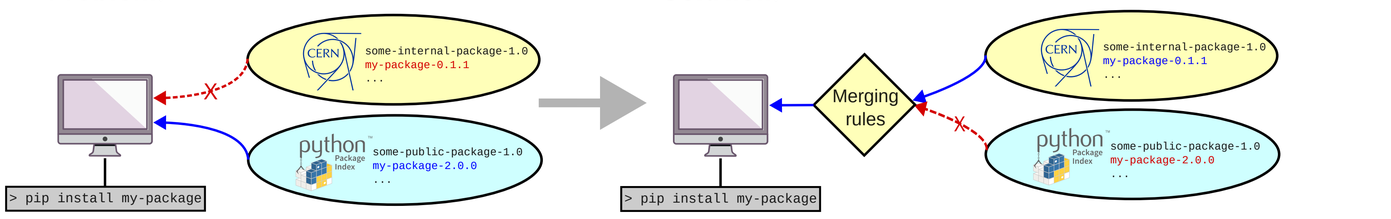
Monitoring and Analytics

- Tracking download and upload actions
- Actions correlated with access and authentication
- Information stored in an ELK stack shared with other services
- Kibana dashboards provide visual analytics capabilities

Security-Oriented

- Eliminates dependency confusion attacks
- Authentication with Gitlab API and in-house Role-Based Access Control (RBAC)
- Per-project ownership
- "Yank" for in-house packages
- "Yank" override for PyPI packages
- Flexibility to introduce new measures

Dependency confusion is a type of supply-chain attack whereby a software installer script is tricked into downloading malicious code from a public repository instead of the intended file with the same name from an internal repository. In other words, **having two repository sources, public and private, poses a security risk**, when the installer has to prioritize the source. Reducing upstream options to a single private feed is one of the possible mitigations that has been implemented here.



Usable by Other Laboratories

The initial prototype has been published on GitHub under an MIT licence and it is hoped that it will trigger interest from other parties that have similar operational needs. With sufficient collaborative interest, there is potential for the project to be openly developed, and to power Python package repositories across many domains.

https://github.com/simple-repository

Contact: ivan.sinkarenko@cern.ch, philip.elson@cern.ch